

A Comparison of Methods for Detecting atypical Trajectories

H. Saul, K. Kozempel, M. Haberjahn

Institute of Transportation Systems, German Aerospace Center, Germany.

Abstract

The detection of atypical trajectories and events in road traffic is a challenging task for the implementation of an intelligent transportation system. It also provides information for optimizing the traffic flow and mitigating risks of accidents without the need to observe individual traffic participants. For detecting such events two methods representing the state of the art are compared: A map-based trajectory analyzer and a neural network, the Self Organizing Map—both applicable with unsupervised learning.

The two compared algorithms detect atypical trajectories by modeling the probability function of trajectory features representing the object state at every trajectory point containing location, speed and acceleration values. The map-based approach was extended and improved by pre-clustering the trajectories with regard to their relation (e.g. vehicle turning left/going straight ahead). The Self Organizing Map algorithm uses vector quantization and prototyping of feature vectors and, thus, does not need any preliminary work. Both methods are evaluated by experiments using the same data which allows strengths and weaknesses to be revealed. The data base for evaluation consists of trajectories from traffic surveillance cameras at an intersection and simulated trajectories.

Keywords: atypical, trajectory, behavior, probability density maps, self organizing map, threshold, pre-clustering, surveillance, automatic, unsupervised.

1 Introduction

Monitoring interesting events that rarely occur in a big volume of data is a tedious and time-consuming job. Thus, computer vision solutions helping to automate the process are needed. Finding and recording these events has a lot of possible applications, for example in helping to analyze the infrastructure, i.e. determining the most common conflict types which can be indicators for infrastructure weaknesses causing conflicts. Another application is recording near-accidents or revealing possible correlations between near-accidents and accidents, what still is a subject of current research.

Due to the fact, that there is no a-priori knowledge for the type of common conflicts and each crossing allows different behaviors—e.g. regarding typical velocities and braking forces—methods are requested that do not need models of the desired events. The challenge is to design a surveillance system that is capable of learning what is normal behavior to distinguish it from abnormal behavior that may lead to conflict situations. The final goal is not the explicit labeling of events but the detection of anomalous, or rare and unlikely events respectively that differ from typical patterns.

In this paper two promising approaches are implemented, improved and compared: A stochastic map-based method called Probability Density Maps (PDM), based on the work of Meysel [1], and the Self Organizing Feature Map (SOFM), based on the work of Owens [2]. Both methods are trajectory-based and model the underlying probability density functions of the data. Moreover, they are applicable in an unsupervised manner. In the following section related work referring to the detection of behavior and movement patterns is presented. In section 2 the two implemented methods are described. Experiments on both methods are conducted in section 3. Subsequently the experiments are evaluated and analyzed. The paper closes with an outlook towards future work.

1.1 Related work

The approaches for detecting uncommon or rare events have often probabilistic character in order to model the frequency distribution of motions and yield a distinction between common and uncommon. Moreover, in this case usually no previous knowledge respectively explicit definitions of the detected motions is necessary. after having collected the data, the system is autonomously able to differ between rare and frequent events.

Furthermore, most approaches in the field of motion pattern detection are based on individual trajectory analysis ignoring interactions between road

users. A lot of research applies Bayesian networks and as a special case Hidden Markov Models (HMM) and different varieties for detecting and predicting behaviors and detecting anomalies (Bashir [3], Vaswani [4] [4], Hervieu [5], Berclaz [6], Nascimento [7], Benezeth [8], Honeng [9]). Several of these publications are semi-supervised and require hand-made activity descriptions. Other publications are limited to certain environments like rooms. Furthermore there exist approaches with the representation of interactions and group activities, as in Oliver [10], Natarajan [11]. The Principle Component Analysis (PCA) is a popular approach as well. Yacoob [12] models effortfully four activities to be recognized (e.g. ‘walking’). Bashir [13] shows a PCA-based trajectory representation in combination with K-MEANS clustering. Other approaches using clustering techniques are given in Chen [14] and Piciarelli [15], applying the Support Vector Machine. Yet, in these publications the capability of adaption over time is not given and the problem of finding appropriate thresholds remains unsolved. More clustering approaches are applied by Anjum [16] using PCA and Mean-Shift and by Hu [17] using fuzzy K-MEANS together with a Bayesian network, both capable of predicting future behaviors and anomaly detection. In addition, artificial neural networks are used in Johnson [18] learning the distribution of flow vectors but, however, no experimental results are presented. Sumpter [19] applies two competitive networks, a symbol and context network, to predict behavior.

2 Methods

2.1 Probability Density Maps

As in Meysel [1] previously developed the trajectories’ properties are described using different maps. These are position, velocity, acceleration and direction. Thus, the 1st of the probability density maps is generated from the count of objects having passed each map point (x, y) (fig. 3a). Red shows more and green less frequented areas. Hence, the map describes an approximated two-dimensional probability density of vehicles’ positions.

The 2nd property to be written into a map is the velocity v distribution of each position (x, y) (fig. 3b). The 3rd map contains the distribution for the objects’ heading angle ang and in the last map the accelerations a are stored (fig. 3c).

In contrast to Meysel [1], not only the average and standard deviation of the properties are saved but a histogram representing the whole distribution. This kind of data provides storing a multimodal distribution (see fig. 1).

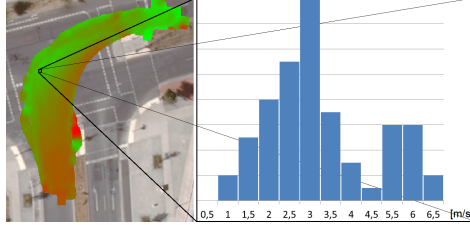


Figure 1: Velocity histogram for one map point

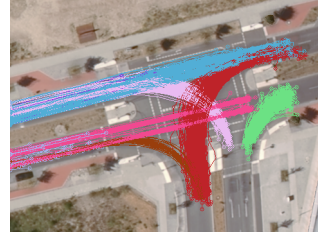
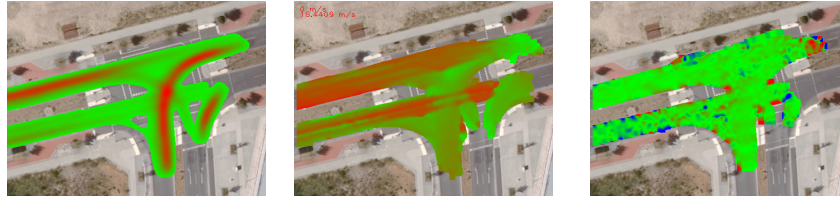


Figure 2: Clustered trajectories



(a) Probability density of vehicles' position (b) Averages of vehicles' velocity (c) Averages of vehicles' acceleration (red) and deceleration (blue)

Figure 3: Components of the PDM

Especially left turning vehicles giving way to the oncoming road traffic are a cause for this.

To have a more dense data basis the histogram bars' height is Gaussian filtered with surrounding map points. The figures 3b and 3c show the Gaussian filtered average velocities and accelerations of motor vehicles while figure 1 visualizes the velocity histogram for one map point.

2.1.1 Trajectory clustering

To distinguish between many different 'typical' trajectories, their start and end positions are clustered with a DBSCAN. Therefore similar paths form clusters (fig. 2). For each of those clusters the four histogram maps (position, velocity, angle, acceleration) are created.

2.1.2 Evaluation

All positions of an incoming trajectory are evaluated with each of the maps. At every position $(x, y)_i$ is compared with the corresponding value $p(x, y)$ in the position map. The more often a position (x, y) was passed during training phase the higher the value is. As the maps are always normalized the value can reach a maximum of 1.0. A trajectory's total position value is computed as the root mean square of all position values. If the aggregated or one of the single normalities is below a certain threshold, the trajectory is marked as unusual.

2.2 Self Organizing Maps

The Self Organizing (Feature) Map, also referred to as Kohonen map, is a two-layer neural network that has been used in a wide field of research topics. The network is able to learn the distribution of data by vector quantization. The SOM is applied in an unsupervised way. The implemented algorithm is based on the work of Owens and Hunter [2] and described in the following. This approach needs no definition of normality. The SOFM estimates data-driven the probability density function of flow vectors and compresses them to 'prototypes' (dimension reduction) serving for comparison with new data. The method consists of two phases: in the 1st stage the feature space gets built by learning from training data and in the 2nd stage the detection can be done by comparing incoming data with the net. The features (see 1) derived from the trajectories and provided as input for the SOM consist of the position x, y in world coordinates, velocity and acceleration:

$$F = (x \ y \ s(x) \ s(y) \ s(dx) \ s(dy) \ s(d^2x) \ s(d^2y))^T \quad (1)$$

with the derivations for the x component calculated by

$$\begin{aligned} dx &= x_t - x_{t-1} \\ d^2x &= x_t - 2x_{t-1} + x_{t-2}. \end{aligned} \quad (2)$$

Owens and Hunter use a sliding window denoted as function s (equation 3) interpolating linearly between two trajectory points in order to smooth the data, but most notably to detect sudden direction changes:

$$s_t(x) = (\mu) (s_{t-1}(x)) + (1 - \mu) (x). \quad (3)$$

The distribution of the data is learnt by adapting the net to the input over several iterations. After the initialisation of the net with random values,

in each iteration t an input feature is picked randomly and the neuron or weight with smallest Euclidian distance to this feature is determined—denoted as Best Matching Unit (BMU). The BMU is the neuron with the smallest distance of all neurons of the net to a given feature vector fulfilling the equation

$$\|x - w_c\| = \min_k \|x - w_k\|. \quad (4)$$

After determining of the BMU, the BMU itself and its neighboring neurons become adapted to the input feature with the adaption rule:

$$w_k(t+1) = w_k + \gamma(t+1) h(t+1, c, k)[x - w_k(t)]. \quad (5)$$

The strength of adaption for every neuron in the net is determined by the neighbourhood function h —typically a two-dimensional Gaussian distribution with its peak at the BMU—and a function γ mitigating the adaption at higher iterations, hence forcing the SOM to converge. The learning phase is finished when the maximum root mean square error (RMSE) between all training data and their BMUs does not decrease significantly over several iterations.

3 Experimental validation and comparison

Both evaluated methods need historic data in order to compile statistics and model the PDF of different properties of trajectories. This data set consists of 2010 trajectories from cars recorded at the same crossing. The test data were recorded and are illustrated in figure 4.

The comparison is based on simulated data. An average trajectory of a left turning car was taken (fig. 5) and manipulated afterwards in different ways in order to simulate several different scenarios. In this way incidents like driving in the wrong direction, hard braking or speeding are simulated. There are avoidance maneuvers as well, simulating a jerky steering movement which is likely to occur for example in situations when a driver mispredicts the behavior of other drivers or was distracted and corrects his path.

1. Original - the original average trajectory
2. Ghost driver - the vehicle passes in opposite direction
3. Small shift - trajectory is shifted 2 m North and 2 m West
4. Big shift - trajectory is shifted 5 m North and 5 m West
5. 2 m/s faster - vehicle drives the same track 2 m/s faster
6. 5 m/s faster - vehicle drives the same track 5 m/s faster
7. 2 m/s slower - vehicle drives the same track 2 m/s more slowly

8. 5 m/s slower - vehicle drives the same track 5 m/s more slowly
9. avoidance maneuver right - vehicle avoids an obstacle to the right
10. avoidance maneuver left - vehicle avoids an obstacle to the left
11. hard braking - vehicle brakes sharply ($-12 m/s^2$) for half a second
12. fast acceleration - vehicle accelerates faster ($12 m/s^2$) for half a second



Figure 4: Training trajectories

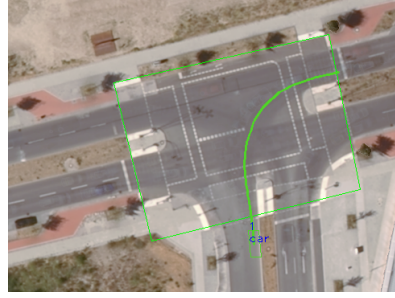


Figure 5: Reference test track

3.1 Probability Density Maps

Original (1) - The original average track (no. 1) is shown in figure 6a. The grey background shows the position distribution with values scaled between black (0) and white (1). Every trajectory position has been evaluated separately in order to show its likelihood. The more green it is the more likely are the position's properties. The average likelihoods for each property are written in the image, while the lowest value for each trajectories is shown in table 1. All four values (p, v, ang, a) lie around 0.8 and the overall is 0.42 which can be considered a very 'normal' trajectory.

Ghost driver (2) - Although position, velocity and acceleration (p, v, a) appear to be totally normal (fig. 6b), the ghost driver's direction (0.001) is conspicuous which makes the whole trajectory an unusual one (0.0006).

Shift (3, 4) - the small shift of 2 m North and West (fig. 6c) results in an unlikely position value of 0.27 and an overall value of 0.08. The big shift of 5 m each yields a very bad position and overall value of 0.03 and 0.004.

Velocity (5, 6, 7, 8) - The faster trajectories (5 and 6), as expected, stand out for their bad velocity likelihoods (0.25 and 0.007) which also result in remarkably low aggregate values (0.12 and 0.004). Although the slower trajectories (7 and 8) would, depending on the threshold, be recognized as

Table 1: Normality values for PDMap and SOFM evaluation

Trajectory type	lowest PDMap value	average SOFM normality
original	0.7317	0.9851
ghost driver	0.0013	0.8907
small shift	0.2716	0.9737
big shift	0.0254	0.9444
2 m/s faster	0.2483	0.9836
5 m/s faster	0.0073	0.9531
2 m/s slower	0.5112	0.9860
5 m/s slower	0.1362	0.9806
avoidance maneuver right	0.3213	0.9669
avoidance maneuver left	0.4340	0.9546
hard braking	0.5630	0.9851
fast acceleration	0.5450	0.9851

well (fig. 6e), their likelihood values are not as distinctive as the faster ones (0.51 and 0.14; 0.25 and 0.07). The 1st one (7) with a velocity difference of only 2 *ms* compared to the original track could even be considered as normal. Left turning vehicles sometimes have to wait for oncoming traffic what makes slow velocities being accepted as normal.

Avoidance Maneuver (9, 10) - Trajectories 9 and 10 do not have one very remarkably low value (lowest is 0.32 and 0.43) but result all together in very low aggregate values (0.09 and 0.12). An additional overall threshold next to one for each single likelihood value seems to be necessary (see fig. 6f).

Braking and Accelerating (11, 12) - The breaking and accelerating trajectories (11 and 12) do not stand out very remarkably (0.24 and 0.23). The acceleration likelihood values (0.63) are only slightly lower than the one from the original trajectory (0.73). Additionally the velocity has a more or less poor value (0.55 and 0.56) which decreases the aggregate likelihood a bit. But nevertheless none of the values is distinctive enough to clearly classify this trajectories as unusual. Although the *a*-values, which should cause the trigger, differ strongly from the histograms, only a twelfth of the whole trajectory contains unusual acceleration values. Furthermore the acceleration is not explicitly estimated in the Extended Kalman Filter, but only

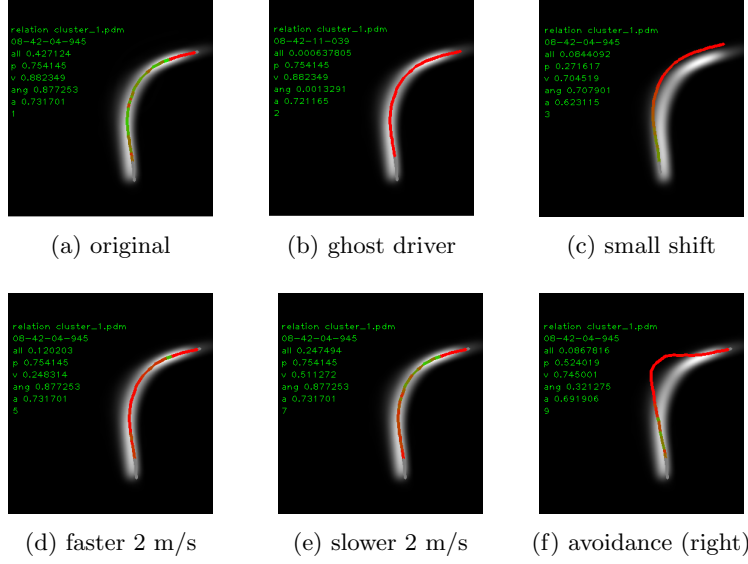


Figure 6: Evaluation of the simulated trajectories by the PDM

generated from velocity differences afterwards. This causes unstable and noise data and sometimes totally implausible values.

3.2 Self Organizing Map

The following experiments have been conducted with a SOFM of net size 20x20 neurons, trained with 2010 trajectories resulting in 147,137 features using $\mu = 0.9$. In Table 1 the evaluation results can be seen. The total normality average value, calculated by averaging the normality of all features belonging to the trajectory, is given. No trajectory point has been classified as abnormal, according to the threshold given by Owens [2]. A trajectory point is classified as normal if the distance to the BMU is below half the maximum RMSE of the training features to the net, to be comparable to the work of [2]. A normality value of 1.0 means no distance of the input feature to the net and 0.0 means the distance is equal or greater than the maximum RMSE.

Original (1) - The normality value of 0.985 as the highest normality value

of all simulated trajectories and being close to 1 suggests this trajectory is detected correctly as normal.

Ghost driver (2) - Since this trajectory differs from the original one only in being in reverse sequence neither the velocity nor the acceleration can be conspicuous. Though the features of the ghost driver's trajectory differ significantly from their BMUs in some of their parts compared to the training trajectories. The average normality is 10% below the original trajectory.

Shift (3, 4) - The shift of 2 m North and West has almost the same average normality as the original trajectory. However the big shift falls below a value of 0.96, as the strongest anomalies like trajectory 6 and 10 do, what indicates this would be an appropriate threshold.

Velocity (5, 6, 7, 8) - The trajectory 2 m/s faster than the original track lies within the range of normality, but the 5 m/s faster trajectory shows significant differences in normality from the original track. This is plausible due to the average velocity of track 6 being above the speed limit of the road. The slower trajectories are not exceptional in their average normality value as expected at cornerings.

Avoidance Maneuver (9, 10) - High abnormality values are expected at positions with sharp steering for the avoidance maneuvers (9 and 10). Due to the fact that the left-turning traffic has to wait and must give way to oncoming traffic the right avoidance maneuver yields better results than the left avoidance. Both have similar results to the trajectory faster by 5 m/s.

Braking and Accelerating (11, 12) - The breaking and accelerating trajectories (11 and 12) have exactly the same classification result as the original track. This is due to the acceleration underlying a big noise.

4 Conclusion and Future Work

It has been shown that both situation detection approaches are able to detect unusual behavior of traffic objects.

The original algorithm of the PDM was improved by pre-clustering the training trajectories. On the one hand the PDMap profits from the automatic clustering step. Therefor motion parameters for certain paths are separated and deviations are more clearly detected (e.g. ghost driver). On the other hand only complete trajectories are accepted for training data. Although slow velocities appear often, the vehicles vanish and cause short trajectories, which do not belong to any cluster. Thus slowly turning vehicles are not weighted sufficiently and are evaluated worse.

Regarding the SOFM, the empirical threshold used by Owen and Hunter [2] for differing between normal and abnormal features could not be confirmed.

Instead a threshold for the total trajectory is suggested. As stated in literature, a challenging problem is finding methodically a reliable threshold for atypical behavior. In the future work we will present a sustainable method for automatic threshold determination. Furthermore the initial training process requires a lot of parameters necessitating to be optimized. This problem might be compensated in online-learning during the surveillance process. The SOFM shows slightly better results in detecting the abrupt avoidances. The avoidance represents only a short part of the trajectory, which is not sufficiently significant to be detected by the PDM. The PDM approach still could be improved by optimizing the weighting of different normality parameters (p, ang, v, a) .

Both approaches have in common to be easily adaptable to new trajectories by inserting them either into histogram maps or the neural net. In this way the methods maintain a relevance to the current situation.

As mentioned above the acceleration values are not yet appropriate to be used for either evaluation method. For forthcoming improvements, a tracking filter (EKF) including acceleration estimation is being developed. In the pending long-term evaluation of data coming from video recordings the adaption capability of both methods will be assessed. Furthermore the detected trajectories will be analyzed in post-processing in order to find out which behavior and parameters exactly led to the trigger.

References

- [1] Meysel, F., *Map-based Multi-Object Dynamic Scene Description and Atypical Event Detection*. Ph.D. thesis, Humboldt Universität zu Berlin, 2008.
- [2] Owens, J. & Hunter, A., Application of the self-organizing map to trajectory classification. *Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, IEEE Computer Society: Washington, DC, USA, pp. 77–, 2000.
- [3] Bashir, F.I., Khokhar, A.A. & Schonfeld, D., Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models. *Image Processing, IEEE Transactions on*, **16**, pp. 1912–1919, 2007.
- [4] Vaswani, N., Chowdhury, A.R. & Chellappa, R., Shape activity: A continuous state hmm for moving/deforming shapes with application to abnormal activity detection, 2005.
- [5] Hervieu, A., Bouthemy, P. & Cadre, J.P.L., A statistical video content recognition method using invariant features on object trajectories.

- IEEE Trans Circuits Syst Video Techn*, **18**, pp. 1533–1543, 2008.
- [6] Berclaz, J., Fleuret, F. & Fua, P., Multi-camera tracking and atypical motion detection with behavioral maps. *ECCV (3)*, pp. 112–125, 2008.
 - [7] Nascimento, J.C., Figueiredo, M.A.T. & Marques, J.S., Trajectory classification using switched dynamical hidden markov models. *IEEE Transactions on Image Processing*, **19**, pp. 1338–1348, 2010.
 - [8] Benezeth, Y., Jodoin, P.M. & Saligrama, V., Abnormality detection using low-level co-occurring events. *Pattern Recogn Lett*, **32**, pp. 423–431, 2011.
 - [9] Hongeng, S. & Nevatia, R., Multi-agent event recognition. *ICCV*, pp. 84–93, 2001.
 - [10] Oliver, N., Rosario, B. & Pentland, A., A bayesian computer vision system for modeling human interactions. *IEEE Trans Pattern Anal Mach Intell*, **22**, pp. 831–843, 2000.
 - [11] Natarajan, P. & Nevatia, R., Coupled hidden semi markov models for activity recognition. *Proceedings of the IEEE Workshop on Motion and Video Computing*, IEEE Computer Society: Washington, DC, USA, WMVC '07, pp. 10–, 2007.
 - [12] Yacoob, Y. & Black, M.J., Parameterized modeling and recognition of activities. *Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, ICCV '98, pp. 120–, 1998.
 - [13] Bashir, F.I., Khokhar, A.A. & Schonfeld, D., Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Transactions on Multimedia*, pp. 58–65, 2007.
 - [14] Xu, Y.C..G.L..K.K.L.Y., Abnormal behavior detection by multi-svm-based bayesian network. pp. 298–303, 2007.
 - [15] Piciarelli, C., Micheloni, C. & Foresti, G.L., Trajectory-based anomalous event detection. *IEEE Trans Circuits Syst Video Techn*, **18**, pp. 1544–1554, 2008.
 - [16] Anjum, N. & Cavallaro, A., Multi-feature object trajectory clustering for video analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, **18(11)**, pp. 1555–1564, 2008.
 - [17] Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T. & Maybank, S.J., A system for learning statistical motion patterns. *IEEE Trans Pattern Anal Mach Intell*, **28**, pp. 1450–1464, 2006.
 - [18] Johnson, N. & Hogg, D., Learning the distribution of object trajectories for event recognition. *Image Vision Comput*, **14**, pp. 609–615, 1996.
 - [19] Sumpter, N. & Bulpitt, A.J., Learning spatio-temporal patterns for predicting object behaviour. *Image Vision Comput*, **18**, pp. 697–704, 2000.